INCREASED PHYSICAL ADDRESSING IN COMPUSERVE MODE

The existing special format with two flag bits and a 16-bit
page number in each halfword becomes a single flag bit
(writable) in bit 0/18, with a 17-bit page number. No Access
is indicated by an all-zero halfword, making a read-only
mapping to physical page 0 impossible.

NEW INSTRUCTIONS FOR USE WITH EXTENDED VIRTUAL ADDRESSING

General:

Several of these instructions have an implied additional
level of indirection in order to permit full extended
addresses. In these cases, C(E) is interpreted as if it
were an indirect word fetched from a nonzero section. Due
to a hardware restriction, the IFIW form (bit 0=1, bit 1=0)
may not correctly handle global indexing when in section 0,
but the EFIW form will operate correctly.

The new stack instructions determine the stack pointer
format in the normal way. Hence, the stack must be local
and mapped in all code sections if it to be used in section
0 at all.

XPUSHJ, opcode 766:

Like PUSHJ, but takes C(E) as a global indirect address
word. Stores PC without flags on stack, even when executed
in section 0.

XPOPJ, opcode 767:

Like POPJ, but interprets the stack word as a full 30-bit
return address, even when executed in section 0.

ZPUSHJ, opcode 770:

Like PUSHJ, but stores PC without flags on stack, even when
executed in section 0. E is normal immediate address.

♀
XXCT, opcode 771:

Equivalent to XCT if E addresses a nonzero section. If E
addresses section 0, it is changed to section 1 after
fetching C(E) but prior to performing   e effective address
calculation for the target instruction, thereby permitting
the target instruction to use extended addressing. The AC

EXTENDED ADDRESS MAPPING IN COMPUSERVE MODE

Mapping now uses section tables located at EPT/UPT 440-477.
These pointers are one word per virtual section. Bit 0 of
each pointer is a valid bit indicating that the section
exists. Bit 1 is set to indicate that pages 340-377 of this
section are to be mapped via UPT locations 400-417 (as in
nonextended TOPS-10 paging). Bits 2-11 are reserved. Bits
12-35 are the physical address of the page table for the
given section, which must be on a half-page boundary. Page
tables are always contiguous half-pages, which can agree
with the former user map format, but not the former exec map
format.

The trap formats for page fail, MUUO, and LUUO are changed
to match the standard extended (not superextended) TOPS-20
formats, in order to accommodate extended addresses.

BITS TO IDENTIFY NEW MAPPING FEATURES

When CompuServe paging is selected, additional bits in the
left half of the "CONI PAG," data reflect the paging
vintage. Mapping need not be enabled to read these bits.
Bit 0 is always set, and bit 1 always clear, in CompuServe
mode. Bit 2 is set if the new page pointer format is
implemented. Bit 3 is set if CompuServe extended addressing
is implemented.

Compatibility Issues with SI Microcode V7

This microcode is completely upward-compatible with version 6.

Changes in SI Channel Microcode V7

The only change from version 6 is that the SCSI reset logic has been
reworked to program around some problems in the NCR53C90B. This should
improve hung recovery where SCSI resets are used, especially in certain
pathological cases which could repetetively hang on each recovery
attempt.

Compatibility Issues with SI Microcode V6

This microcode is generally upward-compatible with version 5, except
that the protocol version has changed to 2. The changes to the
configuration
flags should be transparent to initiator-mode use, and target mode didn't
exist
in version 5.
The ROM version is no longer identical to the disk version, since
the
normal code is now too large for the ROM. Currently the only difference
is
that the ROM version never does quadword memory cycles.
Due to a bug in the way MSP checked the code size against the CRAM
size, this version cannot be loaded by MSP earlier than V156. The
"squeezed"
version used in the ROM does not have this problem.

Changes in SI Channel Microcode V6

1)   Target mode is now implemented, as per the spec.

2)   The request ID field of the RB (word 7) is now propagated to the
      request ID field of the SB.

3)   The Select Enable bit in the configuration flags is now split into
      separate Select Enable and Reselect Enable bits, with the old bit
      being redefined as Reselect Enable for compatibility.

4)   Various bugs relating to resets and aborts have been fixed.

5)   A bug in the internal unit queue logic has been fixed.

6)   A problem with the 53C90B's handling of SCSI reset has been
partially
      programmed around.

7)   The front-end buffer area is now large enough to handle magtape-
size
      blocks.

8)   The host data transfer buf´ ·s have been expanded from 64 to 128
      words, for improved throug___ut (at some expense in latency).

Compatibility Issues with SI Microcode V5

This microcode is fully compatible with version 4. The file
SI10U.NAN
is no longer required, since the LCA data is now included in the .BIN
file.

Changes in SI Channel Microcode V5

1)   Early disconnection by the target no longer causes a microcode
hang.
     In particular, addressing another SI30 as a target no longer causes
     a microcode hang.

2)   Addressing nonexistent buses is now caught correctly, rather than
     causing a variety of nasty things to happen.

3)   The LUN should now be stored in the CDB if the target doesn't
accept
     the IDENTIFY message.

4)   The status ring full test should now be correct.

5)   The synchronous transfer parameters are now renegotiated after
     receiving CHECK CONDITION status, in order to program around
targets
     which fail to renegotiate after power up.

6)   A couple of pathological bugs are fixed.

Compatibility Issues with SI Microcode V4

This microcode is fully compatible with version 3.

Changes in SI Channel Microcode V4

This is a maintenance release to correct a bug in reselection where
the resuming operation transfers data to/from host memory and the
previously
initiated command (e.g. SEEK, MSP buffer operation) did not. This bug
was
also in version 2 but only with respect to front-end operations.

Compatibility Issues with SI Microcode V3

This microcode is compatible with all SI boards implementing the "interrupt" branch condition, which has now been retrofit into the two early prototypes. Previous microcode versions will not correctly handle multiple buses, even when only one bus is configured.

Changes in SI Channel Microcode V3

1) Multiple buses are now fully supported.

2) Write operations now prefetch the first buffer (64 words) in parallel with the initial command interpretation by the drive. This prefetch is wasted if the drive initially disconnects on a write, but this should not normally be the case for an on-cylinder transfer.

3) Target-initiated synchronous transfer negotiation now conforms to the SCSI spec, making it acceptable to the more fussy drives. This eliminates the problem of running in asynchronous mode if a drive is powered down and up with the system running.

4) BUS DEVICE RESET now resets the synchronous transfer parameters to asynchronous (and "needing negotiation"), as per the SCSI spec.

5) Abort functions now force renegotiation of synchronous transfer parameters on the next subsequent command. The current settings are not affected.

6) The "message out retry" mechanism via unsolicited message out phase is now implemented, as per the SCSI spec.

7) An excessively long "wire-or glitch" on BSY, which confuses the 53C90, no longer hangs the microcode, although the affected bus remains hung until a hardware bus reset is given.

8) Non-SI boards plugged into the same channel are no longer poked during LCA configuration.

Compatibility Issues with SI Microcode V2

There are no compatibility problems between this and the V1 microcode, but the accompanying LCA data (SI10U.NAN) contains additional logic required by all SI cards other than the original prototype. This is
fully upward compatible with the previous version and may be used with the old card.

Changes in SI Channel Microcode V2

1)   Numerous bugs discovered in the initial field test have been fixed.

2)   SAVE DATA POINTER is now acceptable at the end, as well as the beginning, of a data transfer.

3)   Buffer channel programs now use 8-byte CCWs, rather than 7, for better alignment. Since MSP does not yet support the SI as a front-end device, this has no operational impact.

4)   The polling code now handles the new dual-bus card correctly, under an assembly conditional. This release of the code has the conditional set for the old mode in order to be compatible with the original prototype SI card.

SI Channel Microcode V1

        This is the initial release of the microcode for the SI SCSI
channel. Its operation mostly conforms to the KL10-SCSI Adapter
Programming Specification.

        Configuration:

        The microcode architecturally supports up to four SI boards
per channel, with up to two SCSI interfaces per board, but the handling
of multiple buses is not quite complete. Buses are addressed as
<board #>*2+<chip #>. Currently only boards 0 and 1 are permitted,
and an interlock against simultaneous use of the host memory data
transfer mechanism by multiple buses is lacking. The current SI
board only provides one bus, but is designed to be expandable to two.
        MSP V150 can recognize an SI interface and set the channel
type appropriately. The automatic channel type is "SI", which emulates
the operation of the KL10-SCSI Adapter. Alternatively, the type can
be manually set to "IUSI", which operates as another "IU10" type
channel, which employs the IU10-style I/O instructions and initial-
ization (similar to SA, FA, SM), but otherwise keeps the same memory
formats as the KL10 version. This is the only mode available if the
CPU is running microcode older than V15. The SI diagnostic can handle
either mode of operation.
        Since the SI board uses a programmable LCA, an additional
file, SI10U.NAN, is required for "microcode" loading. Normally
the LOAD MICROCODE command loads both the control store and the LCA,
although this can be modified with switches. It is not possible to
load the LCA without loading the control store at least once, as
routines in the microcode are required to load the LCA.

        Changes to Programming Specification:

1)   To support multiple SCSI buses, the configuration data in
        the communication page has been expanded to four words.
        Offset 10+n contains information for board n, with the
        chip 0 information in bits 0:15 and the chip 1 information
        in bits 16:31. This is upward-compatible with the KL
        version, which is viewed as having bus 0 only.

2)   Because of the multiple buses and the potential for
        multiple CPUs, it is necessary to indicate which buses
        should actually be set to '   given IDs. The SI channel
        requires that the high-order bit of the byte containing
        the ID (bit 8 or 24 of the word) be set to a 1 for that

a per-bus, per-CPU basis, regardless of the ID enable.

3)   The original format of configuration information provided
       after a RESET (by the KL version) included hardware and
       firmware versions, but not a protocol version.   Since
       two very different devices are attempting to provide the
       same protocol, the firmware version is inappropriate for
       the purpose.   Byte 1 has been redefined as the protocol
       version, beginning with a value of 1.   The hardware
       version has been moved to byte 3.   The default ID has been
       moved from byte 3 to byte 4+n, where n is the bus number.
       The format of the default ID byte(s) has also been changed
       to match the format of the corresponding byte in the
       communication page, i.e., shifted 3 places and with the
       high bit set if the bus and ID is valid.   Note that the
       initial KL version has a hardware version of 0, which
       appears in the new view to be a protocol version of 0,
       thus permitting these formats to be distinguished.   The
       KL version will be changed in the future to conform to
       this format.

4)   The various special CCWs in the communication page which
       are present only to make the KL10 MBOX happy are not
       required by the SI channel, regardless of the emulation
       mode.   The ROP and SIP pointers are still used, but can
       be pure addresses.   All other words given as XFER, XFERH,
       or TIC are ignored.   All pointers are interpreted as
       full 28-bit addresses, but for compatibility, a ROP or
       SIP with bit 0 set is masked to 22 bits and rewritten.
       Software must not compare pointers in a way which depends
       on the opcode/count fields.   Any IUSI channel, or an SI
       channel with bit 8 set in the CONI, can be assumed to
       have the extended addressing capability (which also
       applies to the CCW formats as in the MI channel).
       The ICWA pointer is also unnecessary by the SI.

5)   The timeout feature provided by the KL version is not
       implemented by the SI.   However, the SI can service
       commands from the host during a data transfer, so any
       timeouts can be more appropriately handled by the
       software and the RESET function.

6)   A non-change but possible gotcha is that Selection Enable
       really does affect reselection as well as selection, so
       that disconnection won't w ': if this bit is off.   The
       SI specifies no disconnect⸗ in the IDENTIFY message
       in this case, as well as REJECTing a DISCONNECT if it

7) The SI does not implement "command supersede". Multiple commands issued to a given BUS/ID/LUN are queued. This is a more desirable behavior in general, but is not really useful until a "request ID" mechanism is implemented. Such a mechanism is essential to support a SCSI-2 version, and avoids some other confusion even in the present scheme. It is not possible to cancel an existing command without using some variant of the RESET function, although note that the ABORT version will not actually select the target unless necessary.

8) The SI Initiate command has no preconceived notions about CDB length, so all 12 bytes are provided to the target if requested, regardless of group code. Additional requested bytes beyond the first 12 are supplied as zeroes. This change should be included in a future KL version.

9) The SI only enforces the data transfer direction bit when set to 1. A 1 in this bit prevents the target from storing into the data buffer, but a 0 does not prevent outputting it.

10) BUS DEVICE RESET terminates all commands on the given ID, since it is not LUN-specific. It is also sent regardless of whether any commands are outstanding. These changes should be included in a future KL version.

11) The MBOX errors are mostly unimplemented. In particular, the long/short word count errors are not provided, although the exact amount of data transferred can be determined from the transfer count. Three CCW-related errors can be reported: illegal CCW location as "address parity error" in bit 17, illegal data address as "NXM" in bit 18 (this is sort of correct except that the legal subset of the transfer is not performed), and illegal CCW as "RH20 error" in bit 20.

12) The SI does not yet implement target mode. This will be added in a future release.

SI/SCSI Compatibility Issues

1) The optional messages currently supported are
   DISCONNECT, ABORT, MESSAGE REJECT, NO OPERATION,
   MESSAGE PARITY ERROR, BUS DEVICE RESET, IDENTIFY,
   and SYNCHRONOUS DATA TRANSFER REQUEST.

2) SYNCHRONOUS DATA TRANSFER REQUEST currently always
   sends the desired values for the period and offset,
   even when responding to a target-initiated negotiation.
   This is not strictly according to spec, although it
   probably works with most targets, and in any case
   doesn't matter in the normal case where the initiator
   starts the negotiation. For maximum compatibility,
   this will be fixed in a future release.

3) SAVE DATA POINTER is currently accepted only in the
   trivial case where the current pointer value is zero.
   This was necessary to avoid confusing the Micropolis
   disk, but is probably not actually appropriate.
   Eventually this function will be fully implemented,
   but it may be completely removed before then, since
   a reasonable target shouldn't depend on support for
   superfluous pointer saves.

4) RESTORE POINTERS is currently accepted but may
   not work correctly.

5) DISCONNECT is rejected if reselection is disabled
   or if the target failed to accept the IDENTIFY.

6) Command-related messages (DISCONNECT, RESTORE POINTERS,
   etc.) are rejected while the SI is trying to ABORT
   or BUS DEVICE RESET out of a command in progress.

7) No IDENTIFY is sent preceding a BUS DEVICE RESET,
   since the reset is not LUN-specific.

8) All EXTENDED MESSAGEs are transferred in entirety, as
   determined by the length, before rejecting.

9) Information phases may occur in any sequence, but
   a phase error will be flag' ' if the target fails
   to supply at least one stat__ byte, since there
   would be no other way to determine the validity

is probably rarely useful without SAVE DATA POINTER.

11)   Since the NCR 53C90B does not provide independent enables for selection and reselection, it is not possible to fail to respond to an incoming target selection if reselection is enabled. The chip will autonomously respond, and accept an IDENTIFY message and CDB bytes (in the usual case). The microcode will then summarily disconnect if the selection is unwanted (currently always the case). The initiator may or may not treat this differently from no response at all.